

### **REMARKS/ARGUMENTS**

The office action of August 21, 2003 has been carefully reviewed and these remarks are responsive thereto. Reconsideration and allowance of the instant application are respectfully requested. Claims 1-35 remain pending in this application.

The wording in certain claims has been amended to improve the clarity of the invention.

Claims 1-35 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. patent no. 5,727,155 to Dawson ("Dawson") in view of U.S. patent no. 6,126,328 to Mallory et al. ("Mallory"). Applicants respectfully traverse this rejection.

The action alleges that the combination of Dawson and Mallory results in the invention of independent claims 1, 17 and 32.

#### **Claims 1-16**

Claim 1 as amended calls for the steps of:

providing at the remote peripheral device notification of an event received from a source application at the host computer;  
responsive to a first user input at the remote peripheral device, displaying on the auxiliary display unit information associated with the event;  
forming a query based upon the first user input; and  
responsive to a second user input at the remote peripheral device, executing the query to cause the source application to be launched by the host computer.

According the claim 1 invention, at a remote peripheral device, a notification of an event (e.g., receipt of a message) is provided from a source application (e.g., messaging application) at the host computer. Then, in response to user input (one or more inputs, e.g., navigating through a messaging application), the invention displays information (e.g., a message) on the auxiliary display unit associated with the event. Next, a query is formed based upon the first user input (e.g., navigation query), and responsive to a second user input (e.g., a button actuation of Open on PC), executing the query causing the source application to be launched by the host computer.

The action contends that Dawson describes a host and remote system each including their own display device and a "remote application (360), which maintains a list of display locations

belonging to shared applications. See Fig. 3 & col. 6, lines 33-41.” Beyond the implication that Dawson shows the elements in the preamble of claim 1, the action has failed to identify any correspondence between the claim language and the disclosures of Dawson and Mallory. Regarding Dawson, the action alleges that Dawson shows that a remote system user can “perform modifications and subsequent transmissions to the host computer (200). See col. 8, lines 34-43.” The action however acknowledges “Dawson does not teach forming a query at the remote device such that execution of a query is to cause an application associated with an event to be launched by the host computer.” The action purports that Mallory “teaches ‘set/remote break event’ receiving the name of the remote module and the breakevent to set, in order that the breakevent (193) is properly supplied. See col. 11, lines 34-38 and Fig. 2.” Also, the action avers that Mallory describes “the host call unit (116) initiating the execution of one or more procedures in the host application program in response to the execution unit (114) detecting the instruction indicating a break event or remote procedure call. See the abstract & Fig. 1.” Applicants are left with little, if any, guidance, as to how the above statements in the action apply to the claims. It appears to applicants that the action relies on Dawson for the concept that information from a shared application can be displayed remotely and modifications can be made remotely as well. Apparently, the action relies on Mallory to at least show all the steps of forming a query and executing the query. The combination of Dawson and Mallory, even if proper, does not result in the claim 1 invention.

The combination of Dawson and Mallory does not teach or suggest the claim 1 combination of steps including providing at the remote peripheral device notification of an event received from a source application at the host computer and executing the query to cause *the source application* to be launched by the host computer. Dawson, as the action acknowledges, does not teach executing any query. Mallory does not teach executing the query to cause the source application to be launched. According to Mallory at col. 2, ll. 38-62,

An apparatus and method for controlling execution of partitioned code are disclosed. The apparatus is an embeddable execution engine having a command unit for receiving a command indicating a sequence of instructions to be executed, an execution unit for executing the indicated sequence of instructions and a host call unit. In controlled-execution mode, the execution unit executes instructions until an instruction indicating a break event or a remote procedure call is detected.

In response to detecting the instruction indicating a break event or a remote procedure call, the host call unit initiates execution of one or more call-back procedures provided in a host application program in which the execution engine is embedded. The host application program call-back procedure is executed to initiate execution of a remote host application program, also having an embedded execution engine according to the present invention, to execute a remote sequence of instructions in its embedded execution engine. The host application program call-back procedure then communicates with a manager process to effectuate an input/output interface to the remote host application program. A computer-user interacts with the host call-back procedure through a single user-interface to set or remove break events and to query execution information in both host application programs, thus achieving controlled execution of partitioned program code.

The Mallory invention has been developed with the purpose of aiding the debugging of partitioned execution code in a distributed application program by controlling the starting and stopping of the execution of code and querying execution information, such as stack, variable and machine data. What is clear from Mallory is that the execution unit executes instructions until an instruction indicating a break event (i.e., halting the execution of code) or remote procedure call is detected (i.e., a procedure call made by the client application requiring execution by the server application; applicable in a partitioned code environment). At that juncture, the host unit initiates execution of a call-back procedure to cause execution of a remote sequence of instructions. A user can interact with the call-back procedure by setting or removing break events and querying execution information (i.e., requesting stack, variable and machine data). In contrast to the claim 1 invention however, in Mallory does not execute the source application in response to any query. Rather, execution information may be requested for debugging purposes or a remote procedure may be called. For at least these reasons, the combination of Dawson and Mallory, even if proper, does not result in the claim 1 invention.

Moreover, applicants submit that one skilled in the art would not have been motivated to modify Dawson to include Mallory because both the remote and host systems in Dawson share control of the host system and share applications in an unlocked access mode. The action merely contends "[o]ne would have been motivated in view of the suggestion in Mallory that the host control unit along with the execution unit (114) as configured in Fig. 2 equivalently provide the desired execution of a query and launching of an application by the host computer." Yet Dawson

does not need to form a query since the host and remote systems can share applications and control of the host system. Indeed, Dawson teaches away from such a modification. Hence, applicants submit that the combination of Dawson and Mallory is improper. Thus, for this additional reason, claim 1 is patentably distinct from the combination of Dawson and Mallory.

Claims 2-16, which depend from claim 1, are patentably distinct over the applied art for the same reasons as claim 1, and further in view of the novel features recited therein. For example, claim 4 recites that the event represents receipt of a calendar message, whereas the events described in Mallory are wholly different and merely consist of initial invocation of an execution engine, detection of a break event, detection of a remote procedure call, and receipt of a request to set or remove a break event for a module not being executed. Mallory, col. 5, ll. 42-46. Claim 6 recites that the notification comprises an audio signal. No teaching or suggestion of a notification being an audio signal is contemplated in either Dawson or Mallory. Regarding claim 13, the action baldly contends that it would have been obvious that the event is a news event and the information includes news information. Applicants do not understand how one skilled in the art would even fathom a notification being a news event and displaying news information for use in the system of Mallory which involves controlling execution partitioned code using break events and remote procedure calls. Claim 16 calls for, among other features, displaying on the auxiliary display unit a plurality of first soft labels representing a plurality of applications and responsive to actuation of a first button on the remote peripheral device associated with one of the first soft labels representing the application, displaying on the auxiliary display unit a plurality of second soft labels, each second soft label representing a different function within the application, the second soft labels replacing the first soft labels on the auxiliary display unit. The action relies on Dawson to show these features of claim 16. However, Dawson does not teach or suggest displaying "soft labels" on the auxiliary display unit much less anywhere. Moreover, Dawson does not teach or suggest that a button on the remote peripheral device is associated with the one of the soft labels as claimed.

**Claims 17-31**

Claim 17 as non-substantively amended, calls for, among other features, providing first information in a context at the first input/output device in response to a first user input, forming a

query associated with providing the first information in the context at the first input/output device; and responsive to a second user input, launching an application based on the query to provide second information in the context at a second input/output device.

The action has failed to establish a case of *prima facie* obviousness with respect to claim 17. Instead the action has again failed to identify any correspondence between the claim language and the disclosures of Dawson and Mallory or a coherent explanation as to how the claim language reads on the combination of Dawson and Mallory. The action has merely supplemented the assertions made regarding Dawson and Mallory discussed above with respect to claim 1 with more statements absent any correspondence with the claim language. The action has further averred the following with respect to claim 17.

Mallory teaches procedure supply breakevent (193) writes the break information to input the data pipe (272) and then returns to the local call back interface procedure. See col. 11, lines 34-41 and Fig. 2. Mallory also teaches the RPC manager including a code which, when executed establishes data pipes (272, 274) through which breakevents are communicated to and from the remote host application (202) and requests for execution information are communicated to the remote host application (202). See col. 10, lines 21-26 & Fig. 2.

Applicants have found no teachings or suggestion in Mallory and Dawson including the portions identified by the action, of the claimed steps of forming a query *associated with providing the first information in the context at the first input/output device* and responsive to a second user input, launching an application based on the query to provide second information *in the context at a second input/output device*. As acknowledged by the action, "Dawson does not teach forming a query at the remote device such that execution of a query is to cause an application associated with an event to be launched by the host computer." Thus to show the steps of forming a query and launching an application based on the query, the action seemingly relies on Mallory. Yet Mallory only describes querying execution information (i.e., requesting stack, variable and machine data) when a break event or remote procedure call occurs. No specific query is formed associated with providing the first information in the context. Indeed no recognition can be found in Mallory of the need to maintain a context. Hence, for at least this reason the combination of Dawson and Mallory does not result in the claim 17 invention.

One of the advantages realized by the claim 17 invention is that a user can launch an application based on the query to provide information in the context when desired and to avoid the need to manually replicate the steps required to provide the second information at the host computer. This problem does not exist with the Dawson system. Significantly, according to Dawson at col. 5, lines 38-46,

The shared application(s) run on host system 200. What appears on display device 225 of remote system 220 is a duplicate image of what is displayed on display device 205 of host system 200. This image is transferred to remote system 220 from host system 200. All mouse and keyboard movements on the shared application(s) entered by the user of remote system 220 are executed on host system 200, subject to the access level accorded to remote system 220, as discussed below.

Hence, in an unlocked mode, Dawson provides a mirror image on the remote system display device of what is currently appearing on the host system display. Thus, no motivation exists to build a query and provide information in the same context when the shared application provides information in the same context in real time. Accordingly, Applicants submit that one would not have been motivated to combine Mallory with Dawson to realize the invention of claim 17. For this reason, the combination of Dawson and Mallory is not proper.

Claims 18-31, which ultimately depend from claim 17, are patentably distinct from the proposed combination of Dawson and Mallory for the same reasons as their base claim and further in view of the novel features recited therein. For example, claim 20 calls for the second information including richer content than the first information. Dawson provides a mirror image on the remote system display device of what was appearing on the host system device, thus the information is identical. There is no motivation or incentive to modify the content to be richer in Dawson, Mallory or otherwise. Claims 27 and 28 call for the first and second information to be news information. Neither Dawson nor Mallory contemplate information including news information. Moreover, claim 27 recites that the first information includes a news story headline and only the second information includes the news story, and claim 28 calls for the first information to include an abstract of a news story and only the second information to include a full version of the news story. To the contrary, Dawson does not contemplate anything other than identical information for a shared application to appear on the host and remote display. In

addition, claim 30 recites that the second user input identifies an input/output device type, and responsive to the second user input, determining that the second input/output device type is the closest input/output device to the first input/output device of the identified type. To show this element, the action relies on col. 5, ll. 4-6 of Mallory. Nonetheless Mallory including col. 5, ll. 4-6 as well as Dawson is wholly devoid of any teaching or suggestion of determining that the second input/output device type is the closest input/output device to the first input/output device of the identified type as claimed.

**Claims 32-35**

Claim 32 has been amended for clarification purposes and not for reasons related to patentability. As amended claim 32 calls for storing a sequence of activations input to the auxiliary user interface to obtain a context on the auxiliary interface, building a query from the sequence of activations, the query being understood by the host computer and responsive to a single activation input to the main user interface or the auxiliary interface, executing the query to provide the context on the main user interface.

The action has failed to establish a case of *prima facie* obviousness with respect to claim 32. Rather the action continues to make allegations regarding obviousness of the invention claim 32 without identifying any correspondence between the claim language and the disclosures of Dawson and Mallory. The action has merely supplemented the assertions made regarding Dawson and Mallory discussed above with respect to claim 1 with more statements absent any correspondence with the claim language. The action has further alleged the following with respect to claim 32.

Mallory teaches an execution engine procedure LoadCode including step (310) where a pointer to the first instruction of the code module to be executed (code-ptr) is stored in the context structure (context-ptr). See col. 14, lines 7-22.

Applicants have found no teachings or suggestion in Mallory or Dawson alone, or in combination, of storing a sequence of activations input to an auxiliary user interface to obtain a context. Mallory merely describes a user can supply line numbers, instructions and other information pertaining to a code module as break events. However, these user inputs do not

constitute a sequence of activations. Nor do these inputs obtain a context on the auxiliary user interface.

The action seemingly relies on Mallory to show the claim step of building a query from the sequence of activations. Notwithstanding the action's contention, Mallory, at most, describes querying execution information (i.e., requesting stack, variable and machine data) when a break event or remote procedure call occurs. Notably, querying execution information by requesting stack, variable and machine data does not amount to building a query from a sequence of activations. In view of the above, the combination of Dawson and Mallory, if proper, does not result in the invention of claim 32.

Moreover, one of the advantages realized by the claim 32 invention is that a query can be built from a sequence of activations and executed, in response to a single activation, to provide the context on a main user interface of a host computer without a user having to manually replicate the sequence of activations required to provide the context on the auxiliary user interface. This problem does not exist with the Dawson system because, in an unlocked mode, Dawson provides a mirror image on the remote system display device of what is currently appearing on the host system display. Applicants submit that one would not have been motivated to combine Mallory with Dawson to realize the invention of claim 32. See Dawson col. 5, lines 38-46. Tellingly, no motivation or need exists to build a query and execute the query to provide information in the context when in Dawson the shared application provides information in the same context in real time. For this reason, one would not have been motivated to combine Dawson and Mallory to obtain the claim 32 invention.

Claims 33-35, which ultimately depend from claim 32, are patentably distinct over the art of record for the same reason as claim 32, and further in view of the advantageous features recited therein.



Appln. No.: 10/029,251  
Amendment dated November 4, 2003  
Reply to Office Action of August 21, 2003

**CONCLUSION**

It is believed that no fee is required for this submission. If any fees are required or if an overpayment is made, the Commissioner is authorized to debit or credit our Deposit Account No. 19-0733, accordingly.

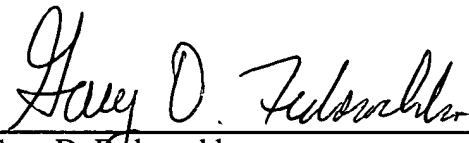
All rejections having been addressed, applicants' respectfully submit that the instant application is in condition for allowance, and respectfully solicit prompt notification of the same.

Respectfully submitted,

BANNER & WITCOFF, LTD.

Dated: November 4, 2003

By:

  
\_\_\_\_\_  
Gary D. Fedorochko  
Registration No. 35,509

1001 G Street, N.W.  
Washington, D.C. 20001-4597  
Tel: (202) 824-3000  
Fax: (202) 824-3001  
GDF:lab